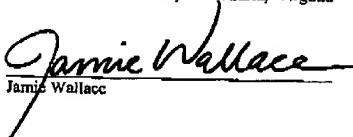


PATENT

MS160309.01/MSFTP170US

CERTIFICATE OF FACSIMILE TRANSMISSION
I hereby certify that this correspondence (along with any paper referred to as being attached or enclosed) is being faxed to 703-872-9306 on the date shown below to Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Date: 5/17/05
Jamie Wallace
Jamie WallaceRECEIVED
CENTRAL FAX CENTER

MAY 17 2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of:

Applicant(s): Kestutis Petiejunas

Examiner: Syed J. Ali

Serial No: 09/730,190

Art Unit: 2127

Filing Date: December 5, 2000

Title: SYSTEM AND METHOD FOR IMPLEMENTING A CLIENT SIDE HTTP STACK

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

05/18/2005 BONNER 02000016 09730190
01 FC:1482

500.00 OP

APPEAL BRIEF

Dear Sir:

Applicants' representative submits this brief in connection with an appeal of the above-identified patent application. A credit card payment form is filed concurrently herewith in connection with all fees due regarding this appeal brief. In the event any additional fees may be due and/or are not covered by the credit card, the Commissioner is authorized to charge such fees to Deposit Account No. 50-1063 [MSFTP170US].

09/730,190MS160309.01/MSFTP170US**I. Real Party in Interest (37 C.F.R. §41.37(c)(1)(i))**

The real party in interest in the present appeal is Microsoft Corporation, the assignee of the present application.

II. Related Appeals and Interferences (37 C.F.R. §41.37(c)(1)(ii))

Appellant, appellant's legal representative, and/or the assignee of the present application are not aware of any appeals or interferences which may be related to, will directly affect, or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. Status of Claims (37 C.F.R. §41.37(c)(1)(iii))

Claims 1-50 stand rejected by the Examiner. The rejection of claims 1-50 is being appealed.

IV. Status of Amendments (37 C.F.R. §41.37(c)(1)(iv))

No claim amendments have been entered after the Final Office Action.

V. Summary of Claimed Subject Matter (37 C.F.R. §41.37(c)(1)(v))**A. Independent Claim 1**

Independent claim 1 recites a client side HTTP stack software component for processing requests, comprising: at least one completion port object; a thread pool comprising a plurality of threads adapted to process tasks associated with at least one client side request; and a client side state machine associated with the at least one request. (See e.g., page 3, line 19-page 5, line 6).

B. Independent Claim 8

Independent claim 8 recites a software component for implementing a client side HTTP stack, comprising: a thread pool comprising N threads adapted to process M requests from a client application component, wherein N and M are integers greater than 1 and wherein M is greater than N. (See e.g., page 6, lines 10-19).

09/730,190MS160309.01/MSFTP170US

C. Independent Claim 23

Independent claim 23 recites a method of implementing a client side HTTP stack, comprising: processing M requests from a client application component using a thread pool comprising N threads, wherein M and N are integers greater than 1 and wherein M is greater than N. (*See e.g.*, page 7, line 5-page 8, line 2).

D. Independent Claim 35

Independent claim 35 recites a computer-readable medium having computer-executable instructions for processing M requests from a client application component using a thread pool comprising N threads, wherein M and N are integers greater than 1 and wherein M is greater than N. (*See e.g.*, page 8, lines 3-8).

E. Independent Claim 46

Independent claim 46 recites a software component for implementing a client side HTTP stack, comprising: means for processing M requests from a client application component using a thread pool comprising N threads, wherein M and N are integers greater than 1 and wherein M is greater than N. (*See e.g.*, page 6, line 10-page 7, line 4).

The aforementioned means for limitations are identified as claim elements subject to the provisions of 35 U.S.C. §112 ¶6. The corresponding structures are identified with reference to the specification and drawings in the parentheticals above corresponding to those claim limitations.

VI. Grounds of Rejection to be Reviewed (37 C.F.R. §41.37(c)(1)(vi))

A. Claim 1 is unpatentable under 35 U.S.C. §102(e) over Sievert *et al.* (US 6,687,729).

B. Claims 8, 23, 35 and 46 are unpatentable under 35 U.S.C. §102(b) over the IBM Technical Disclosure Bulletin ("Control of Dynamic Threads Pool for Concurrent Remote Procedure Calls").

C. Claim 2 is unpatentable under 35 U.S.C. §103(a) over Sievert *et al.* in

09/730,190MS160309.01/MSFTP170US

view of Jones *et al.* (US 6,003,061).

D. Claim 3 is unpatentable under 35 U.S.C. §103(a) over Sievert *et al.* in view of Okano *et al.* (US 6,725,253).

E. Claim 4 is unpatentable under 35 U.S.C. §103(a) over Sievert *et al.* in view of Paxhia *et al.* (US 6,493,749).

F. Claim 5 is unpatentable under 35 U.S.C. §103(a) over Sievert *et al.* in view of Paxhia *et al.* and further in view of Jones *et al.*

G. Claim 6 is unpatentable under 35 U.S.C. §103(a) over Sievert *et al.* in view of Sievert *et al.* in view of Paxhia *et al.* and Jones *et al.* and further in view of Okano *et al.*

H. Claim 7 is unpatentable under 35 U.S.C. §103(a) over Sievert *et al.* in view of Paxhia *et al.* and further in view of Okano *et al.*

I. Claims 9-13, 17-19, 24-28, 32-34, 36-39 and 47 are unpatentable under 35 U.S.C. §103(a) over the IBM Technical Disclosure Bulletin in view of Sievert *et al.*

J. Claims 14, 29, 40 and 48 are unpatentable under 35 U.S.C. §103(a) in view of Sievert *et al.* and Jones *et al.*

K. Claims 15, 30 and 41 are unpatentable under 35 U.S.C. §103(a) over the IBM Technical Disclosure Bulletin in view of Sievert *et al.* in view of Jones *et al.* and further in view of Okano *et al.*

L. Claims 16, 31 and 42-45 are unpatentable under 35 U.S.C. §103(a) over the IBM Technical Disclosure Bulletin in view of Jones *et al.* and Okano *et al.* and further in view of Paxhia *et al.*

M. Claim 20 is unpatentable under 35 U.S.C. §103(a) over the IBM Technical Disclosure Bulletin in view of Jones *et al.*

N. Claim 21 is unpatentable under 35 U.S.C. §103(a) over the IBM Technical Disclosure Bulletin in view of Okano *et al.*

O. Claim 22 is unpatentable under 35 U.S.C. §103(a) over the IBM Technical Disclosure Bulletin in view of Paxhia *et al.*

P. Claim 49 is unpatentable under 35 U.S.C. §103(a) over the IBM Technical Disclosure Bulletin in view of Sievert *et al.* and further in view of Okano *et al.*

Q. Claim 50 is unpatentable under 35 U.S.C. §103(a) over the IBM Technical

09/730,190

MS160309.01/MSFTP170US

Disclosure Bulletin in view of Sievert *et al.* and further in view of Paxhia *et al.*

VII. Argument (37 C.F.R. §41.37(c)(1)(vii))

A. Rejection of Claim 1 under 35 U.S.C. §102(e)

Claim 1 stands rejected under 35 U.S.C. §102(e) as being anticipated by Sievert *et al.* (US 6,687,729). Withdrawal of this rejection is respectfully requested for at least the following reasons. Sievert *et al.* fails to teach or suggest each and every limitation set forth in independent claim 1.

A single prior art reference anticipates a patent claim only if it *expressly or inherently describes each and every limitation set forth in the patent claim*. *Trintec Industries, Inc. v. Top-U.S.A. Corp.*, 295 F.3d 1292, 63 USPQ2d 1597 (Fed. Cir. 2002); *See Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). The *identical invention must be shown in as complete detail as is contained in the ... claim*. *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989) (emphasis added).

Independent claim 1 recites a client side HTTP stack software component for processing requests that comprises at least one completion port, *a thread pool comprising a plurality of threads adapted to process tasks associated with at least one client side request*, and *a client side state machine associated with the at least one request*. Sievert *et al.* does not teach or suggest these novel features of the claimed invention.

Sievert *et al.* discloses a system and method for managing a pool of threads for executing thread operations, and in particular, managing a pool of threads for executing queued items of work. (*See Abstract, and col. 1, lines 16-17*). The Final Office Action (dated October 19, 2004) indicates that the substance of independent claim 1 can be found at col. 3, lines 20-32 and col. 3, lines 34-64. In particular, the Final Office Action asserts that the limitation: *a thread pool comprising a plurality of threads adapted to process tasks associated with at least one client side request*, can be found at col. 3, lines 20-32. Applicant's representative avers to the contrary. Col. 3, lines 20-32 discloses a

09/730,190

MS160309.01/MSFTP170US

work queue, a thread pool, a work queue controller, and a thread pool manager, wherein the work queue supports a method for managing individual items of work in the work queue which may be executed by any of the threads in the thread pool. The cited document however, is silent with respect to the fact that the plurality of threads that comprises the thread pool are adapted to process tasks associated with at least one client side request. Moreover, it should be noted that Sievert *et al.* is directed towards the management of a pool of threads, rather than a client side HTTP stack component that utilizes a thread pool to process client side requests.

In addition, the Final Office Action contends that Sievert *et al.*, at col. 3, lines 34-65, provides *a client side state machine associated with the at least one request*. Applicant's representative respectfully disagrees. Sievert *et al.* at the noted passage provides, a single work queue that can be in one of three states: stopped, suspended and running, rather than a client side state machine associated with at least one client side request. Implicit in the language of the subject claim, on the other hand, is the fact that since the thread pool comprises a plurality of threads, and the plurality of threads is adapted to process tasks associated with at least one client side request, that each client side request must therefore be associated with its own client side state machine. Thus, whereas Sievert *et al.* provides a single work queue that can comprise one of three states to effectuate work to be performed by the multiplicity of threads, applicant's claimed invention provides a client side state machine for each client side request that utilizes a thread within the thread pool.

In view of at least the foregoing, it is submitted that the invention as claimed and the cited document are clearly distinguishable, and that the rejection of independent claim 1 (and claims that depend there from) should be withdrawn.

09/730,190MS160309.01/MSFTP170US

B. Rejection of Claims 8, 23, 35 and 46 Under 35 U.S.C. §102(b)

Claims 8, 23, 35 and 46 stand rejected under 35 U.S.C. §102(b) as being anticipated by IBM Technical Disclosure Bulletin ("Control of Dynamic Threads Pool for Concurrent Remote Procedure Calls"). Reversal of this rejection is respectfully requested for at least the following reasons. The IBM Technical Disclosure Bulletin does not teach or suggest all limitations set forth in the subject claims.

Independent claims 8, 23, 35 and 46 recite similar limitations, namely: a software component for implementing a client side HTTP stack, comprising, a thread pool comprising N threads adapted to process M requests from a client application component, wherein N and M are integers greater than 1 and wherein M is greater than N. It is apparent that the claimed invention utilizes a thread pool created on a client that comprises a multiplicity of threads wherein each thread is utilized by a client application component. The number of threads that comprises the thread pool is set such that the number of threads created in the thread pool is greater than the anticipated number of requests received from the client application component. The IBM Technical Disclosure Bulletin fails to teach or suggest these exemplary aspects of the invention as claimed.

The IBM Technical Disclosure Bulletin discloses an algorithm for controlling and destroying executor threads on an application server, wherein the algorithm creates only a fraction of the threads requested by the application server at initialization time and varies the number of threads when the load on the server changes. It is apparent that the method provided by the IBM Technical Disclosure Bulletin is explicitly confined to an application server, such that threads are created in a thread pool on an application server. This is in contrast to applicant's claimed invention, wherein the thread pool is created on the client rather than on the server. Further, the IBM Technical Document Bulletin discloses that the threads are related to Remote Procedure Calls that are received by the application server from clients accessing the functionality of the application server.

In addition, the Examiner is reminded that the standard by which anticipation is to be measured is *strict identity* between the cited document and the invention as claimed, not mere equivalence or similarity. *See, Richardson* at 9 USPQ2d 1913, 1920. This means that in order to establish anticipation under 35 U.S.C. §102, the single document cited must not only expressly or inherently describe each and every limitation set forth in

09/730,190

MS160309.01/MSFTP170US

the patent claim, but also the identical invention must be shown in as complete detail as is contained in the claim. The fact that the Examiner in the Response to Arguments section of the instant Final Office Action states "there is no reason to believe that the thread pool could not be implemented on the client side" is an explicit concession and tacit acknowledgement that the IBM Technical Document Bulletin not only fails to expressly or inherently describe each and every limitation set forth in the subject claims, but also that the cited document, in the final analysis, does not provide an invention identical to that recited in the subject claims.

In view of at least the foregoing, it is respectfully submitted that the rejection of independent claims 8, 23, 35 and 46, and associated dependent claims, should be reversed.

C. Rejection of Claim 2 Under 35 U.S.C. §103(a)

Claim 2 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Sievert *et al.* in view of Jones *et al.* (US 6,003,061). This rejection should be withdrawn for at least the following reasons. Claim 2 depends from independent claim 1, and for at least the reasons noted *supra*, Jones *et al.* fails to rectify the deficiencies inherent in Sievert *et al.* Accordingly, withdrawal of this rejection is respectfully requested.

D. Rejection of Claim 3 Under 35 U.S.C. §103(a)

Claim 3 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Sievert *et al.* in view of Okano *et al.* (US 6,725,253). Withdrawal of this rejection is requested for at least the following reasons. Claim 3 depends from independent claim 1, and Okano *et al.* fails to makeup the aforementioned deficiencies of Sievert *et al.* regarding claim 1. Accordingly, this rejection should be withdrawn.

E. Rejection of Claim 4 Under 35 U.S.C. §103(a)

Claim 4 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Sievert *et al.* in view of Paxhia *et al.* (US 6,493,749). This rejection should be withdrawn for at least the following reason. Claim 4 depends from independent claim 1; and Paxhia *et al.* fails to rectify the above-noted deficiencies of Sievert *et al.* Withdrawal of this

09/730,190

MS160309.01/MSFTP170US

rejection is respectfully requested.

F. Rejection of Claim 5 Under 35 U.S.C. §103(a)

Claim 5 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Sievert *et al.* in view of Paxhia *et al.* as applied to claim 4 above, and further in view of Jones *et al.* Withdrawal of this rejection is respectfully requested in view of the fact that claim 5 depends from independent claim 1, and Paxhia *et al.* and Jones *et al.* fail to makeup for the aforementioned deficiencies presented in Sievert *et al.*

G. Rejection of Claim 6 Under 35 U.S.C. §103(a)

Claim 6 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Sievert *et al.* in view of Paxhia *et al.* in view of Jones *et al.* as applied to claim 5 above, and further in view of Okano *et al.* Claim 6 depends from independent claim 1, and Paxhia *et al.*, Jones *et al.* and Okano *et al.*, alone or in combination, fail to makeup the aforementioned deficiencies presented by Sievert *et al.* Withdrawal of this rejection is respectfully requested.

H. Rejection of Claim 7 Under 35 U.S.C. §103(a)

Claim 7 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Sievert *et al.* in view of Paxhia *et al.* as applied to claim 4 above, and further in view of Okano *et al.* Reversal of this rejection is respectfully requested for at least the following reasons. Claim 7 depends from independent claim 1, and Paxhia *et al.* and Okano *et al.* fail to cure the aforementioned deficiencies presented by Sievert *et al.* This rejection should be withdrawn.

I. Rejection of Claims 9-13, 17-19, 24-28, 32-34, 36-39 and 47 Under 35 U.S.C. §103(a)

Claims 9-13, 17-19, 24-28, 32-34, 36-39 and 47 stand rejected under 35 U.S.C. §103(a) as being unpatentable over the IBM Technical Disclosure Bulletin in view of Sievert *et al.*. This rejection should be reversed for at least the following reasons.

Claims 9-13, 17-19, 24-28, 32-34, 36-39 and 47 respectively depend from independent

09/730,190

MS160309.01/MSFTP170US

claims 8, 23, 35 and 46, and Sievert *et al.* does not makeup for the aforementioned deficiencies of the IBM Technical Disclosure Bulletin. Accordingly, withdrawal of this rejection is respectfully requested.

J. Rejection of Claims 14, 29, 40 and 48 Under 35 U.S.C. §103(a)

Claims 14, 29, 40 and 48 stand rejected under 35 U.S.C. §103(a) as being unpatentable over the IBM Technical Disclosure Bulletin in view of Sievert *et al.* as applied to claims 13, 28, 39 and 47 above respectively, and further in view of Jones *et al.* Withdrawal of this rejection is respectfully requested for at least the following reasons. Claims 14, 29, 40 and 48 depend from independent claims 8, 23, 35 and 46 respectively, and Sievert *et al.* and Jones *et al.* do not makeup for the aforementioned deficiencies of the IBM Technical Disclosure Bulletin. Thus, it is respectfully submitted that this rejection should be withdrawn.

K. Rejection of Claims 15, 30 and 41 Under 35 U.S.C. §103(a)

Claims 15, 30 and 42 stand rejected under 35 U.S.C. §103(a) as being unpatentable over the IBM Technical Disclosure Bulletin in view of Sievert *et al.* in view of Jones *et al.* as applied to claims 14, 29 and 40 above respectively, and further in view of Okano *et al.* Claim 15, 30 and 41 depend from independent claims 8, 23 and 35 respectively, and the combination of Sievert *et al.*, Jones *et al.* and Okano *et al.* fails to rectify the deficiencies presented in the IBM Technical Disclosure Bulletin. Accordingly, this rejection should be withdrawn.

L. Rejection of Claims 16, 31 and 42-45 Under 35 U.S.C. §103(a)

Claims 16, 31 and 42-45 stand rejected under 35 U.S.C. §103(a) as being unpatentable over IBM Technical Disclosure Bulletin in view of Sievert *et al.* in view of Jones *et al.* and Okano *et al.* as applied to claims 15, 30 and 41 above respectively, and further in view of Paxhia *et al.* Reversal of this rejection is respectfully requested for at least the following reasons. Claims 16, 31 and 42-45 depend from independent claims 8, 23 and 35, and Sievert *et al.*, Jones *et al.* and Okano *et al.* fail to makeup for the deficiencies presented by the IBM Technical Disclosure Bulletin. Accordingly, this

09/730,190

MS160309.01/MSFTP170US

rejection should be reversed.

M. Rejection of Claim 20 Under 35 U.S.C. §103(a)

Claim 20 stands rejected under 35 U.S.C. §103(a) as being unpatentable over the IBM Technical Disclosure Bulletin in view of Jones *et al.* Withdrawal of this rejection is respectfully requested for at least the following reasons. Claim 20 depends from independent claim 8, and Jones *et al.* does not makeup for the deficiencies identified above in the IBM Technical Disclosure Bulletin. Accordingly, withdrawal of this rejection is respectfully requested.

N. Rejection of Claim 21 Under 35 U.S.C. §103(a)

Claim 21 stands rejected under 35 U.S.C. §103(a) as being unpatentable over IBM Technical Disclosure Bulletin in view of Okano *et al.* Claim 21 depends from independent claim 8, and Okano *et al.* fails to rectify the deficiencies presented in the IBM Technical Disclosure Bulletin. Thus, it is submitted that this rejection should be withdrawn.

O. Rejection of Claim 22 Under 35 U.S.C. §103(a)

Claim 22 stands rejected under 35 U.S.C. §103(a) as being unpatentable over the IBM Technical Disclosure Bulletin in view of Paxhia *et al.* This rejection should be withdrawn for at least the following reasons. Claim 22 depends from independent claim 8, and Paxhia *et al.* fails to makeup for the aforementioned deficiencies in the IBM Technical Disclosure Bulletin. Accordingly, reversal of this rejection is respectfully requested.

P. Rejection of Claim 49 Under 35 U.S.C. §103(a)

Claim 49 stands rejected under 35 U.S.C. §103(a) as being unpatentable over the IBM Technical Disclosure Bulletin in view of Sievert *et al.* as applied to claim 47 above, and further in view of Okano *et al.* Reversal of this rejection is respectfully requested for at least the following reasons. Claim 49 depends from independent claim 46, and the combination of Sievert *et al.* and Okano *et al.* fail to rectify the aforementioned

09/730,190

MS160309.01/MSFTP170US

deficiencies presented in the IBM Technical Disclosure Bulletin.

Q. Rejection of Claim 50 Under 35 U.S.C. §103(a)

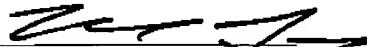
Claim 50 stands rejected under 35 U.S.C. §103(a) as being unpatentable over the IBM Technical Disclosure Bulletin in view of Sievert *et al.* as applied to claim 47 above, and further in view of Paxhia *et al.* This rejection should be withdrawn for at least the following reasons. Claim 50 depends from independent claim 46, and the combination of Sievert *et al.* and Paxhia *et al.* fails to makeup for the aforementioned deficiencies in the IBM Technical Disclosure Bulletin. Thus, it is submitted that this rejection should be reversed.

R. Conclusion

For at least the above reasons, the claims currently under consideration are believed to be patentable over the cited references. Accordingly, it is respectfully requested that the rejections of claims 1-50 be reversed.

If any additional fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Respectfully submitted,
AMIN & TUROCY, LLP


Himanshu S. Amin
Reg. No. 40,894

AMIN & TUROCY, LLP
24th Floor, National City Center
1900 East 9th Street
Telephone: (216) 696-8730
Facsimile: (216) 696-8731

09/730,190

MS160309.01/MSFTP170US

VIII. Claims Appendix (37 C.F.R. §41.37(c)(1)(viii))

1. A client side HTTP stack software component for processing requests, comprising:
 - at least one completion port object;
 - a thread pool comprising a plurality of threads adapted to process tasks associated with at least one client side request; and
 - a client side state machine associated with the at least one request.
2. The client side HTTP stack implementation of claim 1, further comprising a scheduler thread adapted to activate an object scheduled to begin sending requests at a specific time.
3. The client side HTTP stack implementation of claim 1, further comprising a DNS thread adapted to resolve domain names into IP addresses.
4. The client side HTTP stack implementation of claim 1, further comprising a timeout thread with a list of active sockets and timers associated with each socket, and adapted to selectively timeout at least one socket according to at least one timer in the list.
5. The client side HTTP stack implementation of claim 4, further comprising a scheduler thread adapted to activate an object scheduled to begin sending requests at a specific time.
6. The client side HTTP stack implementation of claim 5, further comprising a DNS thread adapted to resolve domain names into IP addresses.
7. The client side HTTP stack implementation of claim 4, further comprising a DNS thread adapted to resolve domain names into IP addresses.

09/730,190MS160309.01/MSFTP170US

8. A software component for implementing a client side HTTP stack, comprising:

a thread pool comprising N threads adapted to process M requests from a client application component, wherein N and M are integers greater than 1 and wherein M is greater than N.

9. The software component of claim 8, further comprising at least one thread activation component adapted to activate at least one of the N threads based on an event.

10. The software component of claim 9, wherein the at least one thread activation component is a completion port.

11. The software component of claim 9, wherein at least one of the N threads is adapted to deactivate itself and return to the thread pool when an operation being processed by the at least one of the threads is pending.

12. The software component of claim 11, wherein the event is the receipt of a completion packet by the at least one thread activation component.

13. The software component of claim 12, wherein the at least one thread activation component is a completion port.

14. The software component of claim 13, further comprising a scheduler thread adapted to activate an object scheduled to begin sending requests at a specific time.

15. The software component of claim 14, further comprising a DNS thread adapted to resolve domain names into IP addresses.

09/730,190MS160309.01/MSFTP170US

16. The software component of claim 15, further comprising a timeout thread with a list of active sockets and timers associated with each socket, and adapted to selectively timeout at least one socket according to at least one timer in the list.

17. The software component of claim 9, further comprising a state machine associated with at least one of the M requests.

18. The software component of claim 17, further comprising at least one key associated with the at least one of the M requests, wherein a first one of the N threads is associated with the at least one of the M requests, and wherein the thread activation component is adapted to associate the context of the first one of the N threads with the at least one state machine using the at least one key, in order to activate the first one of the N threads.

19. The software component of claim 18, wherein the thread activation component is adapted to associate the context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event.

20. The software component of claim 8, further comprising a scheduler thread adapted to activate an object scheduled to begin sending requests at a specific time.

21. The software component of claim 8, further comprising a DNS thread adapted to resolve domain names into IP addresses.

22. The software component of claim 8, further comprising a timeout thread with a list of active sockets and timers associated with each socket, and adapted to selectively timeout at least one socket according to at least one timer in the list.

23. A method of implementing a client side HTTP stack, comprising:

09/730,190MS160309.01/MSFTP170US

processing M requests from a client application component using a thread pool comprising N threads, wherein M and N are integers greater than 1 and wherein M is greater than N.

24. The method of claim 23, further comprising:
selectively deactivating at least one of the N threads; and
activating at least another of the N threads based on an event using at least one thread activation component.

25. The method of claim 24, wherein the at least one thread activation component is a completion port.

26. The method of claim 24, wherein selectively deactivating at least one of the N threads comprises deactivating the at least one of the N threads when an operation being processed by the at least one of the N threads is pending.

27. The method of claim 26, wherein activating at least another of the N threads based on an event comprises:
receiving a completion packet using the thread activation component; and
activating one of the N threads upon receipt of the completion packet using the thread activation component.

28. The method of claim 27, wherein the at least one thread activation component is a completion port.

29. The method of claim 28, further comprising activating an object scheduled to begin sending requests at a specific time using a scheduler thread.

30. The method of claim 29, further comprising resolving domain names into IP addresses using a DNS thread.

09/730,190MS160309.01/MSFTP170US

31. The method of claim 30, further comprising selectively timing out at least one socket according to at least one timer associated with the at least one socket using a timeout thread comprising a list of active sockets and timers associated with each socket.

32. The method of claim 26, further comprising associating a state machine with at least one of the M requests.

33. The method of claim 32, further comprising:
associating at least one key with the at least one of the M requests;
associating a first one of the N threads with the at least one of the M requests; and
associating a context of the first one of the N threads with the at least one state machine using the at least one key, in order to deactivate the first one of the N threads.

34. The method of claim 33, further comprising associating a context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event.

35. A computer-readable medium having computer-executable instructions for processing M requests from a client application component using a thread pool comprising N threads, wherein M and N are integers greater than 1 and wherein M is greater than N.

36. The computer-readable medium of claim 35, further comprising computer-executable instructions for:

selectively deactivating at least one of the N threads; and
activating at least another of the N threads based on an event using at least one thread activation component.

37. The computer-readable medium of claim 36, wherein the at least one thread activation component is a completion port.

09/730,190MS160309.01/MSFTP170US

38. The computer-readable medium of claim 36, wherein the computer-executable instructions for selectively deactivating at least one of the N threads comprises computer-executable instructions for deactivating the at least one of the N threads when an operation being processed by the at least one of the N threads is pending.

39. The computer-readable medium of claim 38, wherein the computer-executable instructions for activating at least another of the N threads based on an event comprises computer-executable instructions for:

receiving a completion packet using the thread activation component; and
activating one of the N threads upon receipt of the completion packet using the thread activation component.

40. The computer-readable medium of claim 39, further comprising computer-executable instructions for activating an object scheduled to begin sending requests at a specific time using a scheduler thread.

41. The computer-readable medium of claim 40, further comprising computer-executable instructions for resolving domain names into IP addresses using a DNS thread.

42. The computer-readable medium of claim 41, further comprising computer-executable instructions for selectively timing out at least one socket according to at least one timer associated with the at least one socket using a timeout thread comprising a list of active sockets and timers associated with each socket.

43. The computer-readable medium of claim 42, further comprising computer-executable instructions for associating a state machine with at least one of the M requests.

44. The computer-readable medium of claim 43, further comprising computer-executable instructions for:

associating at least one key with the at least one of the M requests;

09/730,190MS160309.01/MSFTP170US

associating a first one of the N threads with the at least one of the M requests; and
associating a context of the first one of the N threads with the at least one state
machine using the at least one key, in order to deactivate the first one of the N threads.

45. The computer-readable medium of claim 44, further comprising computer-executable instructions for associating a context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event.

46. A software component for implementing a client side HTTP stack, comprising:

means for processing M requests from a client application component using a thread pool comprising N threads, wherein M and N are integers greater than 1 and wherein M is greater than N.

47. The software component of claim 46, further comprising:

means for selectively deactivating at least one of the N threads; and
means for activating at least another of the N threads based on an event.

48. The software component of claim 47, further comprising means for activating an object scheduled to begin sending requests at a specific time.

49. The software component of claim 47, further comprising means for resolving domain names into IP addresses.

50. The software component of claim 47, further comprising means for selectively timing out at least one socket according to at least one timer associated with the at least one socket.

IX. Evidence Appendix (37 C.F.R. §41.37(c)(1)(ix))

09/730,190

MS160309.01/MSFTP170US

None.

X. Related Proceedings Appendix (37 C.F.R. §41.37(c)(1)(x))

None.